

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего  
образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Лабораторная работа № 2

по программированию

(Вариант 1111)

Студент

Дубинин Артём Сергеевич

группа Р3115

Преподаватель

Карасева Мария Александровна

Санкт - Петербург 2023 год

## СОДЕРЖАНИЕ

1	Задание .....	3
2	Код программы.....	4
3	Результат работы.....	12
4	Заключение.....	15

## 1 Задание

Ознакомиться с документацией, обращая особое внимание на классы `Pokemon` и `Move`. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.

Скачать файл `Pokemon.jar`. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние `jar`-файлы к своей программе.

Написать минимально работающую программу и посмотреть как она работает.

```
Battle b = new Battle();
```

```
Pokemon p1 = new Pokemon("Чужой", 1);
```

```
Pokemon p2 = new Pokemon("Хищник", 1);
```

```
b.addAlly(p1);
```

```
b.addFoe(p2);
```

```
b.go();
```

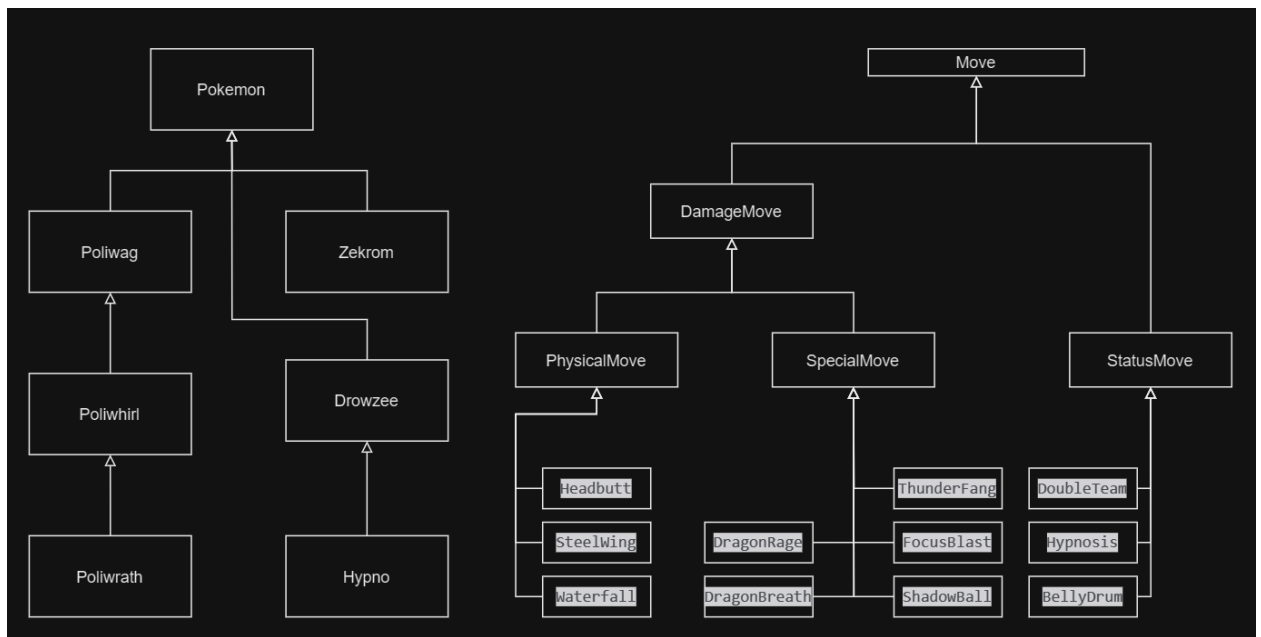
Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса `Pokemon`. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.

Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса `PhysicalMove` или `SpecialMove`. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод `describe`, чтобы выводилось нужное сообщение.

Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники `StatusMove`), скорее всего придется разобраться с классом `Effect`. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.

Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

## Диаграмма классов объектной модели:



## 2 Код программы

Main:

```
import Pokemons.*;
import ru.ifmo.se.pokemon.Battle;
import ru.ifmo.se.pokemon.Pokemon;

public class Main {
    public static void main(String[] args) {
        Battle b = new Battle();
        Zekrom p1 = new Zekrom("Дракончелла", 1);
        Poliwag p2 = new Poliwag("Блюп", 1);
        Poliwirl p3 = new Poliwirl("Блюп X2", 1);
        Drowzee p4 = new Drowzee("Нос > <", 1);
        Hypno p5 = new Hypno("Колдун", 1);
        Poliwraith p6 = new Poliwraith("Ламбада", 1);
        b.addAlly(p2);
        b.addAlly(p1);
        b.addFoe(p4);
        b.addFoe(p3);
        b.addFoe(p5);
        b.addFoe(p6);
        b.go();
    }
}
```

Покемоны:

Drowzee:

```
package Pokemons;

import Moves.Hypnosis;
import Moves.Waterfall;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;
import Moves.DoubleTeam;
```

```

import Moves.ShadowBall;
import Moves.Headbutt;
public class Drowzee extends Pokemon {
    public Drowzee(String name, int level) {
        super(name, level);
        this.setStats(60, 48, 45, 43, 90, 42);
        this.setType(Type.PSYCHIC);
        this.setMove(new DoubleTeam(), new ShadowBall(), new Headbutt());
    }
}

```

Hypno:

```

package Pokemons;

import Moves.DoubleTeam;
import Moves.Headbutt;
import Moves.ShadowBall;
import Moves.FocusBlast;
import ru.ifmo.se.pokemon.Type;

public class Hypno extends Drowzee{
    public Hypno(String name, int level){
        super(name, level);
        this.setStats(85,73,70,73,115,67);
        this.setType(Type.PSYCHIC);
        this.setMove(new DoubleTeam(), new ShadowBall(), new Headbutt(), new
FocusBlast());
    }
}

```

Poliwag:

```

package Pokemons;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;
import Moves.Waterfall;
import Moves.Hypnosis;

public class Poliwag extends Pokemon {
    public Poliwag(String name, int level) {
        super(name, level);
        this.setStats(40, 50, 40, 40, 40, 90);
        this.setType(Type.WATER);
        this.setMove(new Waterfall(), new Hypnosis());
    }
}

```

Poliwhirl:

```

package Pokemons;

import Moves.Hypnosis;
import Moves.Waterfall;
import Moves.BellyDrum;
import ru.ifmo.se.pokemon.Type;

public class Poliwhirl extends Poliwag {

```

```

    public Poliwhirl(String name, int level) {
        super(name, level);
        this.setStats(65, 65, 65, 50, 50, 90);
        this.setType(Type.WATER);
        this.setMove(new Waterfall(), new Hypnosis(), new BellyDrum());
    }
}

```

Poliwrath:

```

package Pokemons;

import Moves.DoubleTeam;
import Moves.Headbutt;
import Moves.ShadowBall;
import Moves.Psychic;
import ru.ifmo.se.pokemon.Type;

public class Poliwrath extends Poliwhirl {
    public Poliwrath(String name, int level) {
        super(name, level);
        this.setStats(65, 65, 65, 50, 50, 90);
        this.setType(Type.WATER);
        this.setMove(new DoubleTeam(), new ShadowBall(), new Headbutt(), new
Psychic());
    }
}

```

Zekrom:

```

package Pokemons;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;
import Moves.ThunderFang;
import Moves.DragonBreath;
import Moves.DragonRage;
import Moves.SteelWing;

public class Zekrom extends Pokemon {
    public Zekrom(String name, int level) {
        super(name, level);
        this.setStats(100, 150, 120, 120, 100, 90);
        this.setType(Type.DRAGON, Type.ELECTRIC);
        this.setMove(new ThunderFang(), new DragonBreath(), new DragonRage(),
new SteelWing());
    }
}

```

Moves:

BellyDrum:

```

package Moves;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;
import ru.ifmo.se.pokemon.Type;

```

```

public class BellyDrum extends StatusMove {

    public BellyDrum() {
        super(Type.NORMAL, 0, 0);
    }

    @Override
    protected String describe() {
        return "BellyDrum";
    }

}

```

DoubleTeam:

```

package Moves;

import ru.ifmo.se.pokemon.*;

public class DoubleTeam extends StatusMove {
    public DoubleTeam() {
        super(Type.NORMAL, 0, 0);
    }

    @Override
    public void applySelfEffects(Pokemon pokemon) {
        pokemon.setMod(Stat.EVASION, +1);
    }

    @Override
    protected String describe() {
        return "Double Team";
    }

}

```

DragonBreath:

```

package Moves;

import ru.ifmo.se.pokemon.*;

public class DragonBreath extends SpecialMove {

    public DragonBreath() {
        super(Type.DRAGON, 60, 100);
    }

    int flag = 0;
    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.3) {
            Effect.paralyze(pokemon);
            flag = 1;
        }
    }

    @Override
    protected String describe() {
        if (flag == 0){
            return "Dragon Breath";
        }
    }

}

```

```

        else
        {
            return "Dragon Breath + PARALYZE";
        }
    }
}

```

Psychic:

```

package Moves;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.Type;

public class Psychic extends SpecialMove {

    public Psychic() {
        super(Type.PSYCHIC, 90, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.1) {
            pokemon.setMod(Stat.SPECIAL_DEFENSE, -1);
        }
    }

    @Override
    protected String describe() {
        return "Псυχтик";
    }
}

```

DragonRage:

```

package Moves;

import ru.ifmo.se.pokemon.*;

public class DragonRage extends SpecialMove {
    public DragonRage() {
        super(Type.DRAGON, 40, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.2) {
            pokemon.setMod(Stat.SPECIAL_DEFENSE, -1);
        }
    }

    @Override
    protected String describe() {
        return "Dragon Rage";
    }
}

```

FocusBlast:



```

package Moves;

import ru.ifmo.se.pokemon.*;

public class FocusBlast extends SpecialMove {
    public FocusBlast() {
        super(Type.FIGHTING, 120, 70);
    }
    int flag = 0;
    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.1) {
            pokemon.setMod(Stat.SPECIAL_DEFENSE, 0);
            flag = 1;
        }
    }
    @Override
    protected String describe() {
        if (flag == 0) {
            return "Focus Blast";
        }
        else {
            return "Focus Blast + SPECIAL_DEFENSE(0)";
        }
    }
}

```

Headbutt:

```

package Moves;

import ru.ifmo.se.pokemon.Effect;
import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Headbutt extends PhysicalMove {
    public Headbutt() {
        super(Type.NORMAL, 70, 100);
    }
    int flag = 0;
    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.3) {
            Effect.flinch(pokemon);
            flag = 1;
        }
    }

    @Override
    protected String describe() {
        if (flag == 0) {
            return "Headbutt";
        }
        else {
            return "Headbutt + FLINCH";
        }
    }
}

```

Hypnosis:

```

package Moves;

import ru.ifmo.se.pokemon.*;
public class Hypnosis extends StatusMove {

    public Hypnosis() {
        super(Type.PSYCHIC, 0, 60);
    }

    @Override
    public void applyOppEffects(Pokemon pokemon) {
        Effect.sleep(pokemon);
    }

    @Override
    protected String describe() {
        return "Hypnosis";
    }
}

```

ShadowBall:

```

package Moves;

import Pokemons.Poliwhirl;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.Type;

public class ShadowBall extends SpecialMove {
    public ShadowBall() {
        super(Type.GHOST, 80, 100);
    }
    int flag = 0;
    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.2) {
            pokemon.setMod(Stat.SPECIAL_DEFENSE, -1);
            flag = 1;
        }
    }

    @Override
    protected String describe() {
        if (flag == 0) {
            return "ShadowBall";
        }
        else {
            return "ShadowBall + SPECIAL_DEFENSE(-1)";
        }
    }
}

```

SteelWing:

```

package Moves;

import ru.ifmo.se.pokemon.*;
public class SteelWing extends PhysicalMove {

    public SteelWing() {
        super(Type.STEEL, 70, 90);
    }
}

```

```

    int flag = 0;
    protected void applySelfEffects(Pokemon pokemon) {
        if (Math.random() <= 0.1) {
            pokemon.setMod(Stat.DEFENSE, +1);
            flag = 1;
        }

    }

    @Override
    protected String describe() {
        if (flag == 0) {
            return "Steel Wing";
        }
        else {
            return "Steel Wing + DEFENSE(+1)";
        }
    }
}

```

ThunderFang:

```

package Moves;

import ru.ifmo.se.pokemon.*;

public class ThunderFang extends SpecialMove {
    public ThunderFang() {
        super(Type.ELECTRIC, 90, 100);
    }

    int flag1 = 0;
    int flag2 = 0;

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        if (Math.random() <= 0.1) {
            Effect.paralyze(pokemon);
            flag1 = 1;
        }
        if (Math.random() <= 0.1) {
            Effect.flinch(pokemon);
            flag2 = 1;
        }
    }

    @Override
    protected String describe() {
        if (flag1 == 0) {
            if (flag2 == 0) {
                return "Thunder Fang";
            } else {
                return "Thunder Fang + FLINCH";
            }
        } else {
            if (flag2 == 0) {
                return "Thunder Fang + PARALYZE";
            } else {
                return "Thunder Fang + PARALYZE + FLINCH";
            }
        }
    }
}

```

```
}  
}
```

Waterfall:

```
package Moves;  
  
import ru.ifmo.se.pokemon.*;  
  
public class Waterfall extends PhysicalMove {  
    public Waterfall() {  
        super(Type.WATER, 80, 100);  
    }  
  
    int flag = 0;  
  
    @Override  
    protected void applyOppEffects(Pokemon pokemon) {  
        if (Math.random() <= 0.2) {  
            Effect.flinch(pokemon);  
            flag = 1;  
        }  
    }  
  
    @Override  
    protected String describe() {  
        if (flag == 0) {  
            return "Waterfall";  
        } else {  
            return "Waterfall + FLINCH";  
        }  
    }  
}
```

### 3 Результат работы

Poliwag Блюп из команды полосатых вступает в бой!

Drowzee Нос >\_< из команды черных вступает в бой!

Poliwag Блюп Waterfall.

Drowzee Нос >\_< теряет 5 здоровья.

Drowzee Нос >\_< ShadowBall.

Poliwag Блюп теряет 4 здоровья.

Poliwag Блюп Waterfall.

Drowzee Нос >\_< теряет 9 здоровья.

Drowzee Нос >\_< теряет сознание.

Poliwhirl Блюп X2 из команды черных вступает в бой!

Poliwhirl Блюп X2 промахивается

Poliwag Блюп Hypnosis.

Poliwhirl Блюп X2 засыпает

Poliwag Блюп Hypnosis.

Poliwhirl Блюп X2 промахивается

Poliwhirl Блюп X2 промахивается

Poliwhirl Блюп X2 Waterfall.

Poliwhirl Блюп X2 теряет 3 здоровья.

Poliwhirl Блюп X2 Waterfall.

Poliwhirl Блюп X2 теряет 3 здоровья.

Poliwag Блюп Hypnosis.

Poliwhirl Блюп X2 засыпает

Poliwhirl Блюп X2 Waterfall.

Poliwag Блюп теряет 3 здоровья.

Poliwag Блюп Waterfall.

Poliwhirl Блюп X2 теряет 3 здоровья.

Poliwag Блюп Waterfall.

Критический удар!

Poliwhirl Блюп X2 теряет 6 здоровья.

Poliwhirl Блюп X2 теряет сознание.

Нурно Колдун из команды черных вступает в бой!

Poliwag Блюп Waterfall.

Нурно Колдун теряет 6 здоровья.

Нурно Колдун ShadowBall.

Poliwag Блюп теряет 7 здоровья.

Poliwag Блюп теряет сознание.

Zekrom Дракончелла из команды полосатых вступает в бой!

Zekrom Дракончелла Steel Wing.

Нурно Колдун теряет 6 здоровья.

Нурно Колдун Focus Blast.

Zekrom Дракончелла теряет 6 здоровья.

Zekrom Дракончелла Dragon Breath.

Нурно Колдун теряет 5 здоровья.

Нурно Колдун парализован

Нурно Колдун теряет сознание.

Poliwrath Ламбада из команды черных вступает в бой!

Poliwrath Ламбада промахивается

Poliwrath Ламбада промахивается

Zekrom Дракончелла Steel Wing.

Poliwrath Ламбада теряет 3 здоровья.

Zekrom Дракончелла увеличивает защиту.

Poliwrath Ламбада Псутчик.

Zekrom Дракончелла теряет 4 здоровья.

Zekrom Дракончелла Dragon Breath + PARALYZE.

Poliwrath Ламбада теряет 9 здоровья.

Poliwrath Ламбада Псутчик.

Zekrom Дракончелла теряет 4 здоровья.

Zekrom Дракончелла теряет сознание.

В команде полосатых не осталось покемонов.

Команда черных побеждает в этом бою!

#### **4 Заключение**

Я разобрался с основными концепциями ООП и научился их использовать на простом примере.